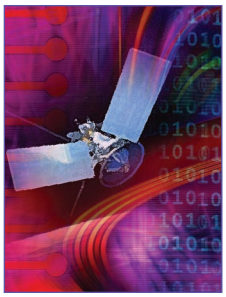

SCTP

New Transport Protocol for TCP/IP

Randall Stewart • *Cisco Systems*
Chris Metz • *Cisco Systems*

The transport layer's primary role is to provide end-to-end communications service between two or more applications running on different hosts. It isolates the applications from the specifics of the underlying network connecting the hosts and provides a simple interface for applications developers. The transport layer can also perform sophisticated actions such as flow control, error recovery, and reliable delivery, which might be necessary for the communicating applications to run properly with reasonable performance.[1]

For the past 20 years, applications and end users of the TCP/IP suite have employed one of two protocols: the transmission control protocol or the user datagram protocol. Yet some applications already require greater functionality than what either TCP or UDP has to offer, and future applications might require even more. To extend transport layer functionality, the Internet Engineering Task Force approved the stream control transmission protocol (SCTP) as a proposed standard in October 2000.[2]

Any application running over TCP can be ported to run over SCTP.

SCTP was spawned from an effort started in the IETF Signaling Transport (Sigtrans) working group to develop a specialized transport protocol for call control signaling in voice-over-IP (VoIP) networks.[3] Recognizing that other applications could use some of the new protocol's capabilities, the IETF now embraces SCTP as a general-purpose transport layer protocol, joining TCP and UDP above the IP layer.

Like TCP, SCTP offers a point-to-point, connection-oriented, reliable delivery transport service for applications communicating over an IP network. It inherits many of the functions developed for TCP over the past two decades, including powerful congestion control and packet loss recovery functions. Indeed, any application running over TCP can be ported to run over SCTP without loss of function, but the many similarities between the two soon give way to several differences. The most interesting of these differences revolve around SCTP's support for multihoming and partial ordering. Multihoming enables an SCTP host to establish a "session" with another SCTP host over multiple interfaces identified by separate IP addresses. Partial ordering lets SCTP provide in-order delivery of one or more related sequences of messages flowing between two hosts. Thus, SCTP can benefit applications that require reliable delivery and fast processing of multiple, unrelated data streams.

TCP Issues

TCP supports the most popular suite of applications on the Internet today, and it has been enhanced in recent years to improve robustness and performance over networks of varying capacities and quality. Nevertheless, it largely retains the behavior outlined in 1981 by Internet pioneer Jon Postel in RFC 793,[4] including properties that make it a less-than-ideal transport protocol for applications such as VoIP signaling or asynchronous transaction-based processing.

TCP requires a strict order-of-transmission delivery service for all data passed between two hosts. This is too confining for applications that can accept per-stream sequential delivery (partial ordering) or no sequential delivery (order-of-arrival delivery).

TCP also treats each data transmission as an unstructured sequence of bytes. It forces applications that process individual messages to insert and track message boundaries within the TCP byte stream. Applications may also need to invoke the TCP push mechanism to ensure timely data transport.

The TCP sockets-based application-programming interface does not support multihoming. An application can only bind a single IP address to a

 1089-7801/01/$10.00 ©2001 IEEE

particular TCP connection with another host. If the interface associated with that IP address goes down, the TCP connection is lost and must be reestablished.

Finally, TCP hosts are susceptible to denial-of-service attacks characterized by TCP SYN "storms" in which a burst of TCP SYN packets arrives to signal an unsuspecting host that the sender wishes to establish a TCP connection with it. The receiving host allocates memory and responds with SYN ACK messages. When the attacker never returns ACK messages to complete the three-way TCP connection setup handshake, the victimized host is left with depleted resources and an inability to service legitimate TCP connection setup requests.[5]

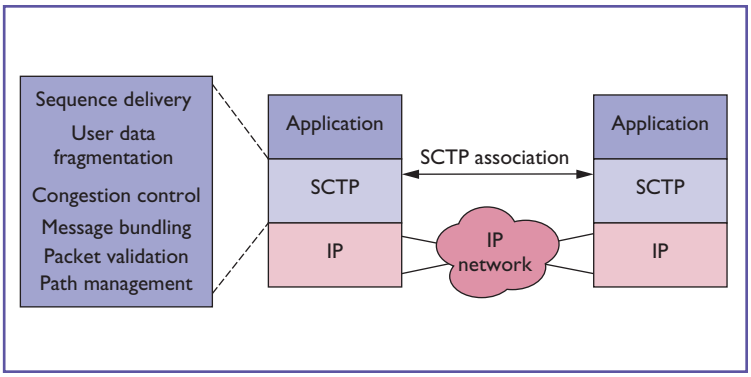

Figure 1. SCTP architecture. SCTP provides enhanced transport layer functionality. Two SCTP hosts form an association employing multiple interfaces to an IP network.

SCTP Features

Figure 1 illustrates SCTP's position within the TCP/IP architecture along with a breakout of its basic functional sublayers. To eliminate the traditional connotation that a "connection" is between a single source and destination address, SCTP uses the term *association* to define the protocol state installed on two peer SCTP hosts exchanging messages. An SCTP association can employ multiple addresses at each end.

SCTP supports some features inherited from TCP and others that provide additional functionality:

- *Message boundary preservation*. SCTP preserves applications' message-framing boundaries by placing messages inside one or more SCTP data structures, called *chunks*. Multiple messages can be bundled into a single chunk, or a large message can be spread across multiple chunks.
- *No "head-of-line" blocking*. SCTP eliminates the head-of-line blocking delay that can occur when a TCP receiver is forced to resequence packets that arrive out of order because of network reordering or packet loss.
- *Multiple delivery modes*. SCTP supports several modes of delivery including strict order-of-transmission (like TCP), partially ordered (per-stream), and unordered delivery (like UDP).
- *Multihoming support*. SCTP sends packets to one destination IP address, but can reroute messages to an alternate if the current IP address becomes unreachable.
- *TCP-friendly congestion control*. SCTP employs the standard techniques pioneered in TCP for congestion control,[6] including slow-start, congestion avoidance, and fast retransmit. SCTP applications can thus receive their share of network resources when coexisting with TCP applications.
- *Selective acknowledgments*. SCTP employs a selective acknowledgment scheme, derived from TCP, for packet loss recovery.[7] The SCTP receiver provides feedback to the sender about which messages to retransmit when any are lost.
- *User data fragmentation*. SCTP will fragment messages to conform to the maximum transmit unit (MTU) size along a particular routed path between communicating hosts. This function is described in RFC 1191 and is optionally employed by TCP/IP to avoid the performance degradation that results when IP routers have to perform fragmentation.[8]
- *Heartbeat keep-alive mechanism*. SCTP sends heartbeat control packets to idle destination addresses that are part of the association. The protocol declares the IP address to be down once it reaches the threshold of unreturned heartbeat acknowledgments.
- *DOS protection*. To mitigate the impact of TCP SYN flooding attacks on a target host, SCTP employs a security "cookie" mechanism during association initialization.

Multihoming

Multihoming probably receives the most attention in discussions of SCTP. It was designed into the protocol to offer network resilience to failed interfaces on the host and faster recovery during network failures. However, the feature's effectiveness is reduced when an end-to-end association path intersects with a single point of failure in the network — a single link or router that all association traffic must pass through, for example, or a host that communicates with only a single interface.

IP networks today are typically resilient to network failure but are often subject to a *reconvergence*

SCTP Resources

Further information on the stream control transmission protocol is available from the following resources.

- University of Delaware Protocol Engineering Lab • http://www.cis.udel.edu/~iyengar/research/SCTP/
- International Engineering Consortium • http://www.iec.org/online/tutorials/sctp/
- Temple University Netlab • http://netlab.cis.temple.edu/SCTP/
- Stream control transmission protocol home • http://sctp.chicago.il.us/sctpoverview.html
- SCTP for Beginners • http://tdrwww.exp-math.uni-essen.de/pages/forschung/sctp_fb/
- *Telecommunications Magazine* picked SCTP as one of the 10 hottest technologies • http://www.telecoms-mag.com/telecom/default.asp?journalid=3&func=articles&page=0105t5&year=2001&month=5
- R. Stewart and Q. Xie, *Stream Control Transmission Protocol (SCTP): A Reference Guide*, Addison Wesley Longman, Boston, Mass., 2001.

time during which the routing network "heals" itself. During this period, traffic can be "black holed" or dropped within the network. Multihomed SCTP end points might be less affected by network reconvergence because lost packets are retransmitted to an alternate address. The SCTP association should thus recover faster and provide better throughput as long as the path to the alternate destination is not affected by the reconvergence.

Stress Reduction

In the quest for network redundancy, enterprises often connect to a second ISP. To ensure that packets can be received over this second link, the customer must advertise a set of addresses (usually obtained from the primary ISP) that fall outside the aggregated address range supported by the second ISP. The second ISP must then advertise its own aggregated address space and customer-specific addresses, resulting in exponential growth in routing table entries.

This practice is unnecessary with SCTP because an association would span the IP addresses contained in the aggregated address ranges supported by both ISPs. SCTP multihoming could therefore be employed to reduce stress on the Internet routing system.

Topology Diversity

SCTP multihoming generally works as designed as long as there is some separation in the routing path of the IP addresses in the association. The diversity in the routing paths dictates the level of fault tolerance an SCTP association experiences. This topology diversity can be physically engineered in small networks, but it is more difficult to achieve in the greater Internet. Some enterprises subscribe to separate tier I and tier II ISPs to optimize their chances of communicating through a separate and diverse network-routing topology. (Most Tier I ISPs forward traffic over their own backbone networks.)

Some argue that the transport layer should remain oblivious to network-layer issues. While other techniques can provide host-interface fault tolerance, however, they might not provide sufficiently fast routing reconvergence for some applications.[9]

Delivery Options

Another area of possible confusion surrounding SCTP is the difference between *reliable* and *ordered* delivery. With TCP, the two are linked in that all data is reliably delivered (lost packets are retransmitted, for example) to the destination host and presented to the application in their transmission sequence. TCP uses a sequence number in each packet's TCP header to perform this task.

SCTP separates the two into independent functions. A transmission sequence number in the SCTP header ensures that all messages are reliably delivered to the destination host, but SCTP has several options in determining which order to present the messages to the destination application. It can use a stream sequence number within the SCTP packet to order messages on a per-stream basis, or it can just kick them up to the application as soon as they arrive. Again, this approach eliminates the head-of-line blocking delay. Note also that TCP behavior can be emulated by placing all messages in a single stream.

SCTP Initiation

As SCTP and TCP are both connection oriented, they require communications state on each host. A TCP connection is defined by two IP addresses and two port numbers. Given two hosts, A and Z, a TCP connection is defined by [IP-A]+[Port-A]+[IP-Z]+[Port-Z] where IP-A and Port-A are one end of the connection and IP-Z and Port-Z are the other.

An SCTP association is defined as [a set of IP addresses at A]+[Port-A]+[a set of IP addresses at Z]+[Port-Z]. Any of the IP addresses on either host

can be used as a source or destination in the IP packet and still properly identify the association.

Before data can be exchanged, the two SCTP hosts must exchange the communications state (including the IP addresses involved) using a four-way handshake, shown in Figure 2. In contrast to TCP's three-way handshake, a four-way handshake eliminates exposure to the aforementioned TCP SYN flooding attacks. The receiver of the initial (INIT) contact message in a four-way handshake does not need to save any state information or allocate any resources. Instead, it responds with an INIT-ACK message, which includes a state cookie that holds all the information needed by the sender of the INIT-ACK to construct its state. The state cookie is digitally signed via a mechanism such as the one outlined in RFC 2104.[10] Both the INIT and INIT-ACK messages include several parameters used in setting up the initial state:

- A list of all IP addresses that will be a part of the association.
- An initial transport sequence number that will be used to reliably transfer data.
- An initiation tag that must be included on every inbound SCTP packet.
- The number of outbound streams that each side is requesting.
- The number of inbound streams that each side is capable of supporting.

After exchanging these messages, the sender of the INIT echoes back the state cookie in the form of a COOKIE-ECHO message that might have user DATA messages bundled onto it as well (subject to path-MTU constraints). Upon receiving the COOKIE-ECHO, the receiver fully reconstructs its state and sends back a COOKIE-ACK message to acknowledge that the setup is complete. This COOKIE-ACK can also bundle user DATA messages with it.

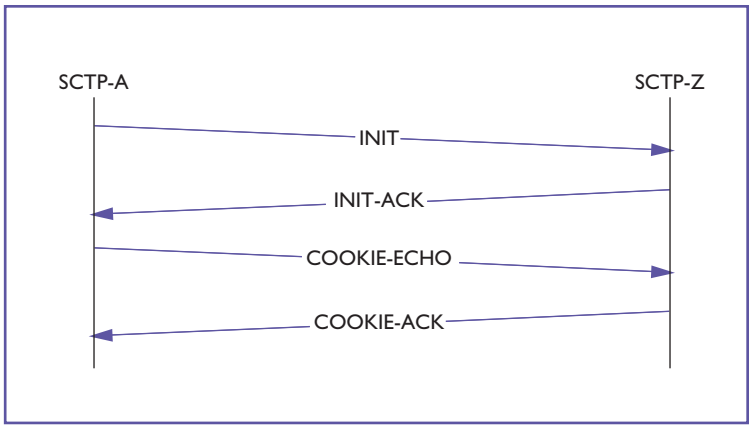

Figure 2. SCTP four-way handshake. This exchange results in the establishment of an SCTP association between two hosts.

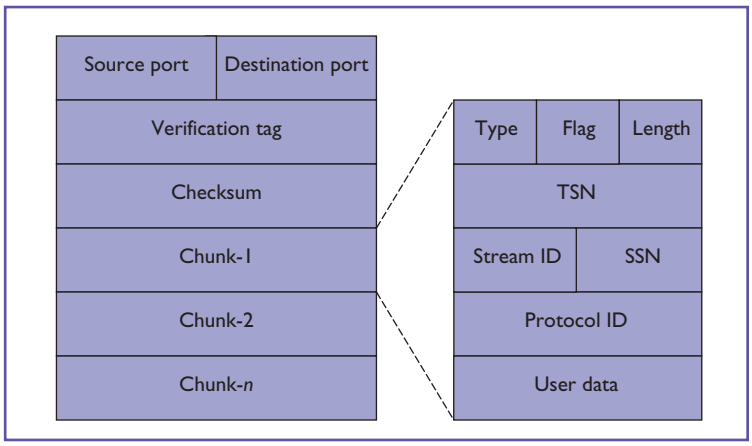

Figure 3. SCTP packet format. A common header (with source and destination port addresses, checksum, and verification tag) is followed by one or more variable-length data chunks.

SCTP Data Transfer

The SCTP message structure facilitates packaging bundled control and data messages in a single format. Figure 3 shows the format of an SCTP packet: A common header is followed by one or more variable-length chunks, which use a type-length-value (TLV) format. Different chunk types are used to carry control or data information inside an SCTP packet.

The SCTP common header contains

- *Source and destination port addresses* that are used with the source and destination IP addresses to identify the recipient of the SCTP packet;
- *Checksum value* to assure data integrity while the packet transits an IP network; and
- *Verification tag* that holds the value of the initiation tag first exchanged during the handshake. Any SCTP packet in an association that does not include this tag will be dropped on arrival. The verification tag protects against old, stale packets arriving from a previous association, as well as various "man-in-the-middle" attacks, and it obviates the need for TCP's timed-wait state, which consumes resources and limits the number of total connections a host can accommodate.

Every chunk type includes TLV header information that contains the chunk type, delivery processing flags, and a length field. In addition, a DATA chunk will precede user payload information with the transport sequence number (TSN), stream iden-

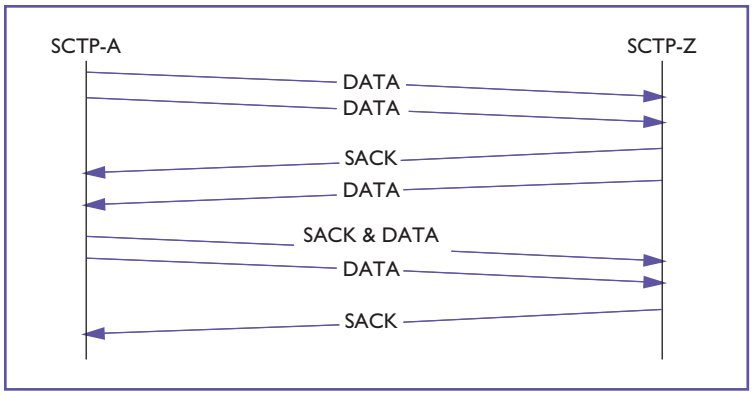

Figure 4. SCTP message exchange. SCTP data and SACK chunks are exchanged between communicating hosts.

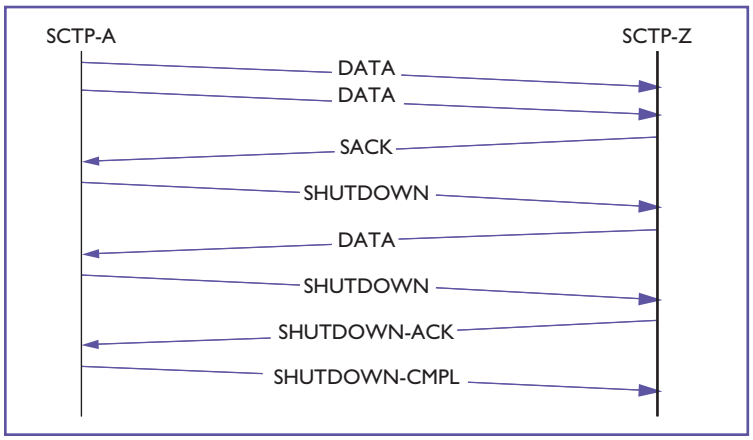

Figure 5. SCTP shutdown. An SCTP association is gracefully terminated by the exchange of a sequence of SHUTDOWN control chunks.

tifier, stream sequence number (SSN), and payload protocol identifier.

The TSN and SSN provide two separate sequence numbers on every DATA chunk. The TSN is used for per-association reliability and the SSN is for per-stream ordering. The stream identifier marks individual messages within the same stream.

Figure 4 shows an example of a normal data exchange between two SCTP hosts. An SCTP host sends selective acknowledgments (SACK chunks) in response to every other SCTP packet carrying DATA chunks. The SACK fully describes the receiver's state, so that the sender can make retransmission decisions based on what has been received. SCTP supports fast retransmit and time-out retransmission algorithms similar to those in TCP.

With few exceptions, most chunk types can be bundled together in one SCTP packet. (SACKs often get bundled during two-way exchanges of user data.) One bundling restriction is that control chunks must be placed ahead of any DATA chunks in the packet.

SCTP Shutdown

A connection-oriented transport protocol needs a graceful method for shutting down an association. SCTP uses a three-way handshake with one difference from the one used in TCP: A TCP end point can engage the shutdown procedure while keeping the connection open and receiving new data from the peer. SCTP does not support this "half closed" state, which means that both sides are prohibited from sending new data by their upper layer once a graceful shutdown sequence is initiated.

Figure 5 depicts a typical graceful shutdown sequence in SCTP. In this example, the application in host A wishes to shut down and terminate the association with host Z. SCTP enters the SHUTDOWN_PENDING state in which it will accept no data from the application but will still send new data that is queued for transmission to host Z. After acknowledging all queued data, host A sends a SHUTDOWN chunk and enters the SHUTDOWN_SENT state.

Upon receiving the SHUTDOWN chunk, host Z notifies its upper layer, stops accepting new data from it, and enters the SHUTDOWN_RECEIVED state. Z transmits any remaining data to A, which follows with subsequent SHUTDOWN chunks that inform Z of the data's arrival and reaffirm that the association is shutting down. Once it acknowledges all queued data on host Z, host A sends a subsequent SHUTDOWN-ACK chunk, followed by a SHUTDOWN-COMPLETE chunk that completes the association shutdown.

SCTP Deployments

There is significant momentum behind developing and deploying SCTP. Nineteen companies, including Ericsson, Motorola, IBM, Cisco, and Nokia, participated in the third SCTP "bakeoff" in April 2001. Operating system support for SCTP that was present included Linux, AIX, Solaris, Windows, and FreeBSD. The success of interoperability tests between various implementations suggests that SCTP will soon find its way into commercial products.

In fact, SCTP code is already available from several parties. Intellinet (http://www.intellinet-tech.com/), a provider of SS7/IP convergence solutions, offers an SCTP protocol stack. Networking protocol software vendor, Data Connection (http://www.dataconnection.com/sctp/), has developed a portable SCTP implementation. Linux kernal source code of SCTP is available from OpenSS7 (http://www.openss7.org/). Several universities are working on SCTP protocol stacks, including Temple and the University of Delaware, and Randall

Stewart, one of the coauthors of this article, has developed a reference implementation under FreeBSD (http://www.sctp.org/).

SCTP continues to gain attention across the standards front. In addition to the ongoing efforts in the Sigtrans and Transport Area working groups, for example, the IETF is actively investigating using SCTP for transport layer support in solutions such as HTTP for enhanced multistreaming Web browsing and Diameter for handling large volumes of AAA messages. Beyond these potential uses, however, SCTP will make its largest initial impact in the VoIP signaling arena.

As IP networks begin to handle more voice traffic, they will need to interwork with telephony networks that use the Signaling System 7 reliable message-based signaling protocol to establish voice circuits. The Sigtrans working group's efforts revolve around adapting and encapsulating SS7 protocol messages in IP packets. Gateways that interface with SS7 and IP networks are prime candidates for establishing SCTP associations with other SS7/IP gateways or VoIP signaling nodes for transporting or backhauling call-setup messages.[11]

This also offers a mechanism for off-loading the native SS7 network with a high-capacity IP packet transport. (The current SS7 network uses relatively low-speed links [56 Kbps] to transport call control messages.) As the mobile Internet begins to take off, this additional capacity will likely be necessary for handling the increase in signaling message volumes introduced by ubiquitous applications such as short message service.

Future Issues

The IETF is currently working on the next revision of the SCTP protocol, which will include several enhancements. For example, Jonathan Stone has shown that a corrupted packet could exit SCTP with a valid Adler-32 checksum (originally used in SCTP) and cause problems at the application layer. Thus, the Transport Area WG will replace the checksum value in the common header with CRC-32, which is superior for handling small packet sizes.

To obtain parity with current operational practices in native SS7 networks, the working group is also studying how to dynamically add or delete IP addresses in an existing association. This enhancement would allow administrators to dynamically add a network interface card (thus a new IP address) to a device (such as an SS7/IP gateway) without having to restart the SCTP association.

With IPv6 back on the horizon, SCTP also needs to work with IPv6 addresses. Knowing how to scope the IPv6 addresses — which ones to list to a peer — becomes a critical issue because SCTP exchanges address lists during association setup. Certain address types supported by IPv6 are not routable (that is, link-local) or reachable outside of specific domains (that is, site-local). If a peer lists an IPv6 site-local or link-local address to a peer that has no connectivity to that address, an association could self-destruct and create a black hole effect.

Work remains to ensure that SCTP is flexible enough to support all the requirements of the next generation of applications, but it is already set to expand transport-layer possibilities beyond what TCP or UDP can offer now.

References

1. P. Amer, S. Iren, and P. Conrad, "The Transport Layer: Tutorial and Survey," *ACM Computing Surveys*, vol. 31, no. 4, Dec. 1999.
2. J. Postel, "Transmission Control Protocol," IETF RFC 793, Sept. 1981; available at http://ietf.org/rfc/rfc793.txt.
3. IETF Signaling Transport working group charter, http://ietf.org/html.charters/sigtran-charter.html.
4. R. Stewart et al., "Stream Control Transmission Protocol," IETF RFC 2960, Oct. 2000; http://ietf.org/rfc/rfc2960.txt.
5. "Defining Strategies to Protect against TCP SYN Denial of Service Attacks," Cisco Systems, tech. memo, Aug. 2001; http://www.cisco.com/warp/public/707/4.html.
6. M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," IETF RFC 2581, Apr. 1999; http://ietf.org/rfc/rfc2581.txt.
7. M. Mathis et al., "TCP Selective Acknowledgment Options," IETF RFC 2018, Oct. 1996; http://ietf.org/rfc/rfc2018.txt.
8. J. Mogul and S. Deering, "Path MTU Discovery," IETF RFC 1191, Nov. 1999; http://ietf.org/rfc/rfc1191.txt.
9. J. Touch, "Dynamic Internet Overlay Deployment and Management Using the X-Bone," *Computer Networks*, July 2001, pp. 117-135.
10. H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," IETF RFC 2104, Feb. 1997; http://ietf.org/rfc/rfc2104.txt.
11. A. Jungmaier et al., "SCTP: A Multi-link End-to-End Protocol for IP-based Networks," *Int'l J. Electronics and Comm.*, vol. 55, no. 1, Jan. 2001.

Randall Stewart is a technical leader with Cisco Systems where he is focusing on wireless Internet technologies and call-control signaling. He is the primary inventor and author of RFC 2960, which defines SCTP. Stewart is coauthor of *Stream Control Transmission Protocol (SCTP): A Reference Guide* (Addison Wesley Longman, 2001).

Chris Metz is a technical leader in the Service Provider Engineering group for Cisco Systems. He is coauthor of *ATM and Multiprotocol Networking* (McGraw-Hill, 1997) and author of *IP Switching: Protocols and Architectures* (McGraw-Hill, 1999). Metz is a member of ACM/Sigcomm and the IEEE.

Readers can contact the authors at {rrs,chmetz}@cisco.com.